TELKOMNIKA, Vol.17, No.2, April 2019, pp.745~752
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i2.11775

Wi-Fi password stealing program using USB rubber ducky

Hansen Edrick Harianto*[1], Dennis Gunawan[2]
Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia
*Corresponding author, e-mail: hansen.edrick@student.umn.ac.id[1], dennis.gunawan@umn.ac.id[2]

Abstract

A minute is all it takes for a hacker to gain informations from your computer, such as Wi-Fi password. Due to the limited capability of people to remember a lot of complex and unique password, people tend to use the same password for most of their account. This paper aimed to implement Wi-Fi password stealing program in USB Rubber Ducky using USB Rubber Ducky Scripting, Visual Basic Script, Web Server, Command Prompt, and Ducky Toolkit to obtain clear text Wi-Fi password that ever connected to the computer. In the testing phase, the success rate of Wi-Fi password stealing program reached 94.28% with 87.87% obtained personal password is still categorized as guessable password and the password reuse rate reached 81.81%. Thus, Wi-Fi password stealing program can be very dangerous as most of the personal password was used in lots of account and still categorized as guessable.

Keywords*: password stealing, password strength, password reuse, USB rubber ducky, WI-FI*

1. Introduction

Along with the rapid improvement of internet of things, computer security is considered as an important aspect because it can harm a lot of people if an individual or an organization successfully found a vulnerability and exploit the system [1-3]. Most of the times, attackers will try to crack their victim's password because users tend to use same password for every web application since users are having trouble remembering a lot of unique and complex password [4-6]. Password is used to protect personal informations as it was selected by the user with easily memorable and not easily guessed by others thus the attackers will try to steal the password using several password stealing attacks, such as phising attack, password stealing program attack, and shoulder surfing attack [7-9].

By leaving your computer unlocked while you are away for seconds, it can give hackers all the time they need to obtain your personal information from your computer. The USB interface is generally a dangerous vector for attack due to their potential for being used as a hacking tool. For example, a USB flash drive can be used to register itself as a device or a keyboard, allowing the attacker to inject malicious scripts inside the computer and it is undetectable [10]. This functionality is present in the Rubber Ducky penetration testing tool [11-13].

As internet becomes a daily need for people nowadays, Wi-Fi is implemented to grant internet access toward the society, allowing electronic devices to communicate without physically connected [14-15]. It is rapidly growing from the aspect of security and accessibility that even one can make their own personal hotspot using their computer, such as desktop, laptop, tablet, and smartphone [16-17]. Most of the hotspot require a password for a machine to connect to the hotspot. Unfortunately, Windows machines can generate the Wi-Fi password in a form of clear text which make it vulnerable toward password stealing attack [18-19]. This is a problem that can be exploited by attackers.

In this paper, the approach details in implementing the penetration testing into Windows machine via USB Rubber Ducky and evaluating the success rate of the program, the strength of victim's personal password, and the password reuse rate will be presented. The mechanism allows the attacker to retrieve all of the victim's Wi-Fi password that ever connected to the victim's machine including their personal Wi-Fi password. This approach will utilize several tools and technologies, such as command prompt, scripting language, web server, and ducky toolkit.

Received May 31, 2018; Revised October 24, 2018; Accepted November 10, 2018

USB Rubber Ducky Infrastructure

USB Rubber Ducky consists of a lot of component and designed to run a malicious script written by the attacker [20]. Components of the USB Rubber Ducky shown in the Figure 1, are:

- Micro SD Storage
- Replay Button
- LED Indicator
- Type a Plug
- 60 Mhz 32-Bit CPU
- Covert Case
- Optional Decal

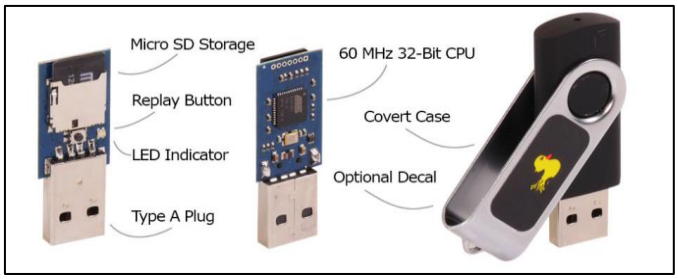

Figure 1. Components of USB Rubber Ducky

USB Rubber Ducky Scripting Language

USB Rubber Ducky use a very simple language and most of the language is based on the keyboard [20]. Some special commands that USB Rubber Ducky have, are:

- REM: to give a comment in the script,
- WINDOWS: has the same function as Windows button on the keyboard,
- DELAY: to postpone an activity that will be run after another activity,
- STRING: to write a sentence into the computer,
- ENTER: has the same function as Enter Button on the keyboard.

All of it can be utilized to create an activity inside the computer. For example, if the attacker wants to execute run program then it can be done using “WINDOWS R” in the script.

Related Works

Based on Benjamin Cannoles experiment, some implementation are already being applied to attack the victim using USB Rubber Ducky. One of them is to install a malware inside the computer using a malicious web server and the capability of USB Rubber Ducky to download the malware and upload sensitive informations back to the server [21]. Another example related to USB Rubber Ducky is an act of stealing Windows logon password using Mimikatz remotely. It can be done by using the capability of USB Rubber Ducky to create a connection to the malicious web server that have mimikatz.exe and sekurlsa.dll. After the connection has been built, the attacker can utilize USB Rubber Ducky to download sekurlsa.dll and run mimikatz.exe remotely [11]. All the implementation of USB Rubber Ducky is undetectable by antivirus because it is considered safe by the computer when someone is typing on their keyboard [22].

2. Research Method

To steal Wi-Fi passwords inside a computer, a script will be written in USB Rubber Ducky language and it will be injected to the USB Rubber Ducky. The script will be divided into 3 main steps based on how system hacking works [23] as illustrated in Figure 2 as follows:

- Gaining access
- Executing program
- Clearing logs

Figure 2. Main flowchart

2.1. Gaining Access

In this step, the attacker will try to get physical access into the computer to run Wi-Fi password stealing program. Ideally, the target will be computers that are left behind by the owner to do a small task that will take less than 5 minutes and it is still running. After the attacker gain physical access into the computer and inject the USB Rubber Ducky into the computer, the script will perform gaining access method as illustrated in Figure 3. The gaining access method consists of opening command prompt as an administrator and deactivating the firewall to perform escalating privileges.

The reason why command prompt should be run as an administrator is because certain commands need administrator privilege to be executed, such as deactivating the firewall. Deactivating the firewall is needed in order to avoid the protection from the firewall to certain features, such as File Transfer Protocol (FTP). Without the protection of the firewall, all kinds of act related to the internet will not be filtered and it is easier for the attacker to proceed to the next method. Deactivation of the firewall will utilize netsh command in the command prompt which is “netsh firewall set opmode mode=disable”.

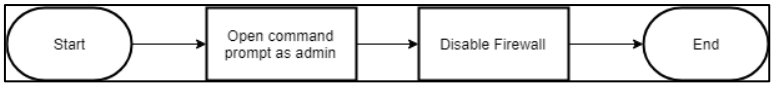

Figure 3. Gaining access module flowchart

2.2. Executing Program

In a big picture, the executing program method consists of creating and changing directory to wifi, extracting wifi informations, creating and executing VB script to compress and zip files, and sending the information file to the attacker server via FTP as illustrated in Figure 4.

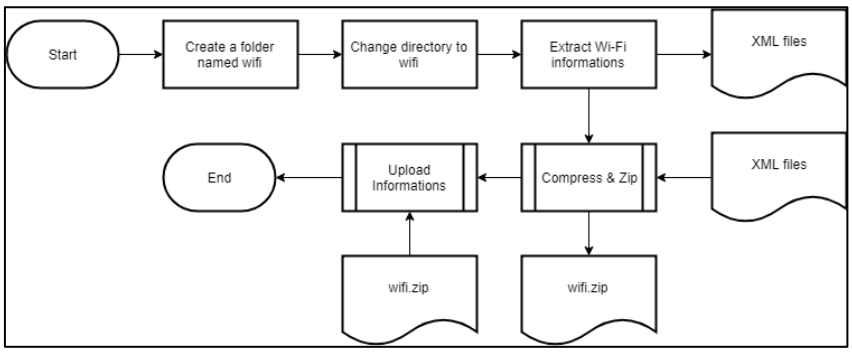

Figure 4. Executing program module flowchart

Wifi directory is created to limit the workspace which will make it easier for the attacker to clear the logs later. Wi-Fi password stealing program will run its main activity using netsh command in the command prompt which is “netsh wlan export profile key=clear”. This command will extract all Wi-Fi informations that ever connected to the computer and export it into Extensible Markup Language (XML) files. After that, the program will create a VB script in order to compress and zip the XML files into one compressed file. VB script was chosen because it has been installed by default in every desktop release of Microsoft Windows. To compress and zip a folder, the VB script will create an empty zipped file and put the source folder inside the empty zipped file as illustrated in Figure 5.

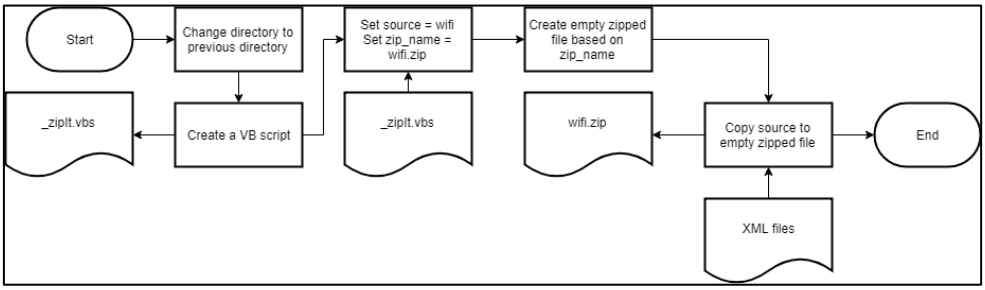

Figure 5. Compress & zip module flowchart

When the file is ready to be sent, upload informations module that illustrated in Figure 6 will be executed. The script will open a connection to the server via FTP and the file will be uploaded to the server using passive mode and binary transfer method. Since it is a compressed file, binary transfer method is used instead of ASCII transfer method because binary transfer method will transfer file as a binary data instead of a text file. The reason why we use passive mode instead of active is to initiates both connections to the server. It solved the problem of firewalls filtering the incoming data port connection to the client from the server.

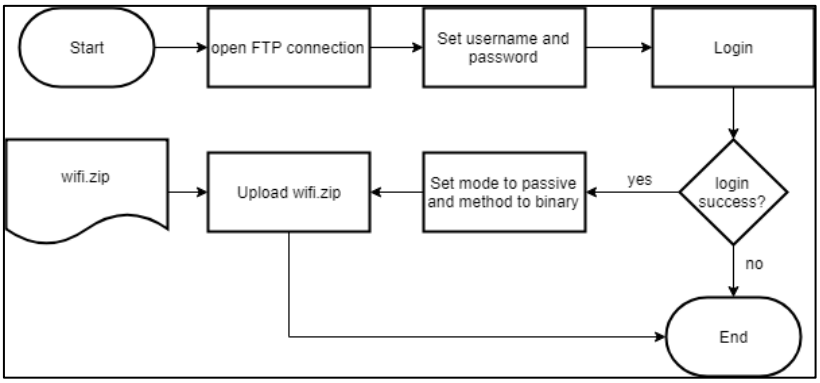

Figure 6. Upload informations module flowchart

2.3. Clearing Logs

In the last step, the script will focus on clearing every single track in the computer as if nothing ever happened to the computer as illustrated in Figure 7 and it consists of closing FTP connection, deleting wifi folder, deleting VB script, deleting wifi.zip, enabling firewall, and closing the command prompt. Based on how system hacking methods work, the application model was illustrated in Figure 8.

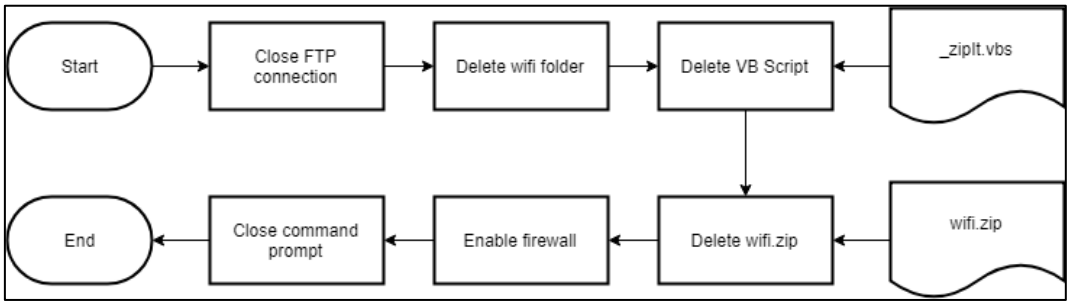

Figure 7. Clearing logs module flowchart

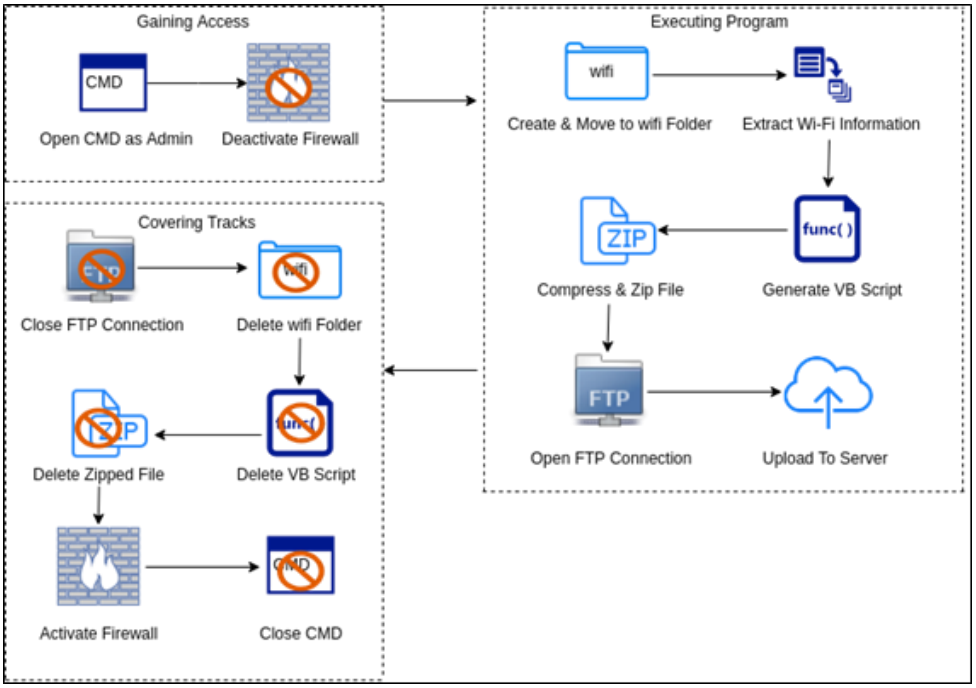

Figure 8. Application model

2.4. Zxcvbn Password Analysis Library

Zxcvbn is one of the popular library to determine a password strength and used by Dropbox [24-25]. Zxcvbn has 3 main process which are match, score, and search. In the match process, Zxcvbn will do pattern matching toward 6 aspects as follows:

- Token
- Reversed
- Sequence
- Repeat
- Keyboard
- Date
- Bruteforce

In the score process, Zxcvbn will do the math based on the match process and determine the password strength by returning a value that indicate the category as shown in Table 1.

Table 1. Zxcvbn Password Strength Category

Value	Category
0	Too guessable
1	Very guessable
2	Somewhat guessable
3	Safely unguessable
4	Very unguessable

Lastly, Zxcvbn will do the search process to estimate the time to guess the password assuming the attacker already knows the structure of the password and this process is based on the result of the match process.

2.5. Evaluation

The data will be collected using voluntary sampling and clustering sampling which means the attacker will ask for the victim consent to collect the data inside the victim's computer

using Wi-Fi password stealing program by injecting USB Rubber Ducky inside the computer. After that, the data will be clustered into 3 category as follows:

- Businessman
- College Student
- Others

The data will be evaluated to determine the success rate of Wi-Fi password stealing program using USB Rubber Ducky, the strength of victim's password using Zxcvbn Password Analysis Library, and the password reuse rate.

3. Results and Analysis

The program was made using USB Rubber Ducky language for Windows operating system and it will run approximately 50 seconds to grab all the Wi-Fi informations in a computer. The result of each Wi-Fi information will be converted into XML file as shown in Figure 9. From this figure, it shows a complete diagnosis of each Wi-Fi which ever connected to the computer including SSID name and the security key for the Wi-Fi. In this testing, 35 samples was used to evaluate the program which consist of 5 businessmen, 20 college students, and 10 others.

```
<?xml version="1.0"?>
<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">
        <name>CEH-702</name>
        <SSIDConfig>
                <SSID>
                        <hex>4345482D373032</hex>
                        <name>       </name>
                </SSID>
        </SSIDConfig>
        <connectionType>ESS</connectionType>
        <connectionMode>auto</connectionMode>
        <MSM>
                <security>
                        <authEncryption>
                                <authentication>WPA2PSK</authentication>
                                <encryption>AES</encryption>
                                <useOneX>false</useOneX>
                        </authEncryption>
                        <sharedKey>
                                <keyType>passPhrase</keyType>
                                <protected>false</protected>
                                <keyMaterial>                    </keyMaterial>
                        </sharedKey>
                </security>
        </MSM>
        <MacRandomization xmlns="http://www.microsoft.com/networking/WLAN/profile/v3">
                <enableRandomization>false</enableRandomization>
                <randomizationSeed>2538524312</randomizationSeed>
        </MacRandomization>
</WLANProfile>
```

Figure 9. Wi-Fi information XML file

3.1. The Success Rate of Wi-Fi Password Stealing Program

The test was executed 35 times and failed 2 times under college student category due to the Wi-Fi SSID contains a character other than numeric and alphabet. In other words, the success rate of Wi-Fi password stealing program reach 94.28% and each test was executed on various computer specification within 1 minute.

3.2. The Strength of Victim's Personal Password

From 33 successful samples, the strength of victim's personal password is shown in Figure 10. Those which categorized as guessable password reach 87.87% and only 12.13% of them was categorized as unguessable. The Strength of Victim's Password Chart is shown in Figure 10.

It shows how most of our personal password is still weak as it only fulfill certain regular expression shown in Table 2 where 48.48% of the samples matched [a-z] regular expression, 18.18% of the samples matched [0-9] regular expression, 27.27% of the samples matched

[a-z0-9] regular expression, and 3.03% of the samples matched [a-zA-Z] and [A-Z0-9] regular expression.

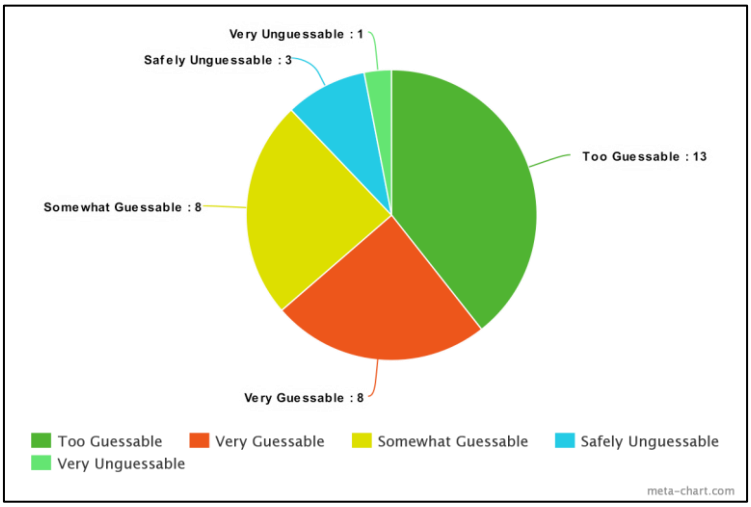

Figure 10. The strength of victim's password chart

Table 2. Regular Expression of Password Samples

Regular Expression	Number of Matched Password
[0-9]	6
[a-z]	16
[a-z0-9]	9
[a-zA-Z]	1
[A-Z0-9]	1

3.3. The Password Reuse Rate

Based on the survery to the victims as illustrated in Figure 11, it is found that 81.81% use the same password for every account which they have, 12.12% use a similar password for every account but with additional characters depends on the website, and only 6.06% of them that use different password for every account. The Password Reuse Rate Chart is shown in Figure 11.

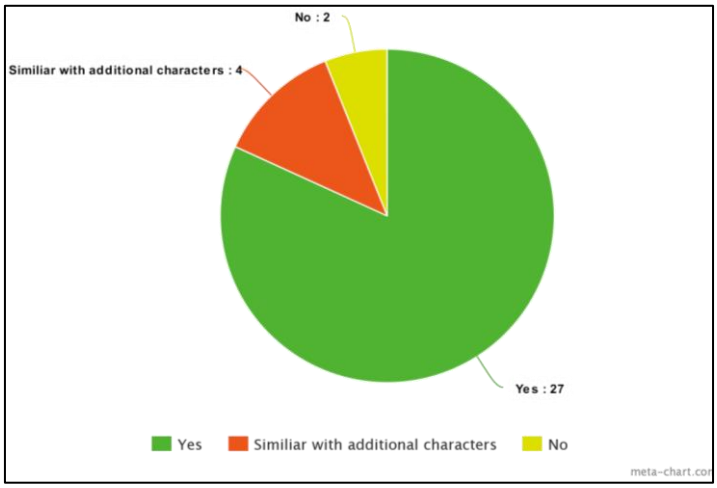

Figure 11. The Password reuse rate chart

4. Conclusion and Future Works

The proposed Wi-Fi password stealing program was implemented successfully with the rate of 94.28% and because of the high rate of password reuse that reach 81.81%, Wi-Fi password stealing can be very dangerous as the password is used in lots of account. How low the password strength is also becomes a consideration as the attacker has a high chance of success to do a brute force attack that fulfill [a-z] regular expression which have 48.48% number of matched. It shows how people are still lacking of awareness in creating a strong password.

This project can be extended in several ways:

- Due to the failed experiment on 2 subjects because of non-alphanumeric character, extend the VB Script to make it successful.
- Instead of using FTP, one can change the firmware of the USB Rubber Ducky and save the files locally inside the USB Rubber Ducky.
- More experiment can be done with different operating system, such as Linux and Mac OS.

References

[1] Phong Chiem Trieu. A Study of Penetration Testing Tools and Approaches. 2014: 7.
[2] Patil Sonali, Jangra Ankur, Bhale Mandar, Raina Akshay, Kulkarni Pratik. Ethical Hacking: The Need for Cyber Security. *IEEE.* 2017: 1602.
[3] Alsunbul Saad, Le Phu, Tan Jefferson, Srinivasan Bala. A Network Defense System for Detecting and Preventing Potential Hacking Attempts. *IEEE*. 2016: 449.
[4] David Silver, Suman Jana, Eric Chen, Collin Jackson, Dan Boneh. Password Managers: Attacks and Defenses. 2014: 1-2.
[5] Bharti Vijay Nikose, Ravindra Gupta, Gaurav Shrivastav. Preventing Password Reuse Attack Using Authentication Protocol. *IJRDET*. 2014; 2(2): 31.
[6] S Al-Sharif, F Iqbal, T Baker, A Khattack. White-hat Hacking Framework for Promoting Security Awareness. *IEEE*. 2016: 1.
[7] Venkadesh S, K Palanivel. A Survey on Password Stealing Attacks and It's Protecting Mechanism. *IJETT*. 2015; 19(4): 223.
[8] Sharayu A. Aghav, RajneeshKaur Bedi. Authentication Mechanism for Resistance to Password Stealing and Reuse Attack. 2014: 1.
[9] S. Megala Devi, M. Geetha. Opass: Attractive Presentation of User Authentication Protocol with Resist to Password Reuse Attacks. *IJCSMC*. 2013; 2(8): 174-175.
[10] Bhakte Rajbhoosan, Zavarsky Pavol, Butakov Sergey. Security Controls for Monitored Use of USB Devices based on the NIST Risk Management Framework. *IEEE*. 2016: 461.
[11] Benjamin Cannoles, Ahmad Ghafarian. Hacking Experiment by Using USB Rubber Ducky Scripting. *IMCIC*. 2017.
[12] Dinesh Mothi. Obfuscating Live Computer Forensic Investigative Process on a Windows 7 Operating System: A Criminal's Perspective. *IJCA*. 2015; 122(6): 3.
[13] Dr. Sunil Sikka, Utpal Srivastva, Rashika Sharma. A Review of Detection of USB Malware. *IJESC.* 2017; 7(7): 14283-14284.
[14] Wadhwa Utkarsh. Wireless Network Security: Tough Times. *IEEE*. 2015: 1022.
[15] Wang Dong, Zhou Ming. A Framework to Test Reliability and Security of Wi-Fi Device. *IEEE.* 2014: 953.
[16] Khoula Al Harthy, Shah Nazaraf, Shankarappa Arun N.S. Smartphone's Hotspot Security Issues and Challenges. *ICITST*. 2016: 113.
[17] Radivilova Tamara, Hassan Hassan Ali. Test for Penetration in Wi-Fi Network: Attacks on WPA2-PSK and WPA2-Enterprise. *IEEE*. 2017: 1.
[18] Jane Butler, Ermanno Pietrosemoli, Marco Zennaro, Carlo Fonda, Stephen Okay, Corinna "Elektra" Aichele, Sebastian Buettrich, Jim Forster, Klass Wierenga, Bruce Baikie, Laura Hosman, Michael Ginguld, Emmanuel Togo. Wireless Networking in the Developing World. Third Edition. 2013: 126-133.
[19] Renuka Muppavarapu. Open Wi-Fi Hotspots–Threat and Mitigations. 2014: 1-3.
[20] Tyler Dever. USB Rubber Ducky Analysis. 2015: 15-16.
[21] Karsten Nohl, Sascha Krißler, Jakob Lell. BadUSB–on Accessories that Turn Evil. 2015: 3-6.
[22] Young, N., Dress, R. Cyber Security for Automatic Test Equipment. *IEEE*. 2018; 21(4): 4.
[23] EC-Council. CEHv9 Module 05 System Hacking. 2016: 5-6.
[24] Wang Ding, He Debiao, Wang Ping, Haibo Cheng. FuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars. *IEEE*. 2016: 595.
[25] Xavier de Carné de Carnavalet, Mohammad Mannan. From Very Weak to Very Strong: Analyzing Password-Strength Meters. 2014: 5-8.